Prerequisites

Courses:
- CSE 219 or 260: Advanced Programming in an HLL
- CSE 220: Systems-Level Programming
- CSE 303: Automata Theory

Programming Experience:
- Significant programming experience in multiple languages.
- Ability (and a willingness) to adapt to new languages and tool chains.

Course Outcomes

At the end of the course, students are expected to have:
- An ability to use of formal attributed grammars for specifying the syntax and semantics of programming languages.
- Working knowledge of the major phases of compilation, particularly lexical analysis, parsing, semantic analysis, and code generation.
- An ability to design and implement a significant portion of a compiler for a language chosen by the instructor.

Informal Statement of Course Objective: To learn the process of translating a modern high-level language to executable code.
- Learn the fundamental techniques from lectures, text book and exercises from the book.
- Apply these techniques in practice to construct a fully working compiler for a non-trivial object-oriented language.
Organization

- Concepts and basic ideas in the lectures.
- Concrete implementation in a series of programming homework assignments.

Grading

<table>
<thead>
<tr>
<th>Component</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homework assignments</td>
<td>30%</td>
</tr>
<tr>
<td>Informal project presentation</td>
<td>5%</td>
</tr>
<tr>
<td>Two mid-term exams</td>
<td>40%</td>
</tr>
<tr>
<td>Comprehensive final exam</td>
<td>25%</td>
</tr>
</tbody>
</table>

Letter grades:

- 80% or above: definite A, provided:
  - at least 5 non-trivial homework submissions
  - 70% or better in the exams
- 50% or above: definite C+ or better provided:
  - at least 3 non-trivial homework submissions
  - 35% or better in the exams
The Rules of the Game

Programming assignments:

- May work in pairs.
- Late submissions (submitted up to 3 days late) will be graded normally, but have the following maximum score limit:

<table>
<thead>
<tr>
<th>Late Days</th>
<th>Max.</th>
</tr>
</thead>
<tbody>
<tr>
<td>On time</td>
<td>100%</td>
</tr>
<tr>
<td>1 day late</td>
<td>75%</td>
</tr>
<tr>
<td>2 days late</td>
<td>50%</td>
</tr>
<tr>
<td>3 days late</td>
<td>25%</td>
</tr>
<tr>
<td>≥ 4 days late</td>
<td>0%</td>
</tr>
</tbody>
</table>

- No late penalty (up to 3 days) for ONE assignment of your choice! You should send me mail ahead of the submission deadline if you want to use this penalty-free late days for any assignment.

Your Responsibilities

- Limit discussion of homework assignments to problems, not solutions.
- Cheating, illegal collaboration and plagiarism will be treated with maximum seriousness.
  - Grad students will be referred to the Graduate Program Director.
  - Undergrad. students will be referred to CASA.

Conviction of academic dishonesty carries substantial penalties, such as receiving an ‘F’ grade, and/or expulsion from the University. See syllabus on course web page for details.
Syllabus

The Importance of Being Earnest

The Importance of Being Earnest

Compiler Design  Course Organization  CSE 504  11 / 19

The Importance of Being Earnest
Effort Level

The effort required by this course is **High**

*but so are the rewards:*

- Hands on experience in large-scale programming (this semester: in Python).
- In-depth knowledge of *how* programs written in high-level languages are translated and executed.

---

Course Structure

- Compilers in this course will be written in **Python**.
  - Python is one of the easiest languages to pick up. I will assume you have sufficient ability to become effective in Python quickly.
- You will write a series of modules that will finally compose to become a full-fledged compiler for an fairly complex Object-Oriented language.
Python

Python is deceptively easy-to-learn programming language that has
- Imperative features: local and global variables and functions.
- Data structures: transparent support for lists, dictionaries, etc.
- Objects: with a dynamic class system
- Higher-order functions: support for programming in a functional style
- Automatic memory management
- Tons of useful libraries
- Syntax that is a bit unusual, but one that makes sense as you get used to the language.

Python can be (and has been) used for a wide range of programming tasks from scripting to numerical packages.

Instructional Support

All course material, including homework handouts, notes, etc. will be posted on the course web site.

We will use the Piazza for
- Course Announcements (check these regularly)
- Course Discussion Board

We will use Blackboard for
- Feedback and Grades on assignments

Assignments will be submitted via git.
Course Discussion Board

- Use this to discuss any course-related material: lectures, homework problems, exams, etc.
- If you have any questions on the material, first check to see if any one else had the same question as you have, and whether the question has been answered already; otherwise post the question on the board.
- We’ll try to answer all questions on the board asap: within 24 hrs for normal days, and much quicker near exam/homework deadlines.

Questions

How to contact course staff:
- Staff contact information is on the web page/Piazza/Blackboard.
- Post your question on the discussion board.
- Meet me during my office hours (or fix an appointment).
- Send me email. (Post on discussion board unless the question is personal.)
- Contact TA (to be announced) by visiting him/her during office hours or by email.
- Grading related questions: send email to the TA (who grades your homework assignments) first; if unresolved, send me email.
Immediate Action Required

- Install Python on your machine if you don’t already have it. We will use Python 2 in this course; **not Python 3**.
- Read HW preparation instructions on course web page (under Homeworks) and send me email to set up your Git repository.